スツルム・ランチョス法 (数値計算のアルゴリズムの研究)

AUTHOR(S):

村田, 健郎; 梅谷, 征雄

CITATION:

村田, 健郎 ...[et al]. スツルム・ランチョス法 (数値計算のアルゴリズムの研究). 数理解析研究所講究録 1982, 453: 89-101

ISSUE DATE:

1982-02

URL:

http://hdl.handle.net/2433/102991

数理解析研究所講究録
453巻 1982年 89-101

スツルム・ランチョス法

村田 健郎 （日立中研）
梅谷 征雄 （ 仝 上 ）

1 はじめに：

最近，ランチョス法系統の二つの論文：

Parlett·Scott の hanczos with Selective Orth. (1979, Math. Comp.)[2]

Ericcson·Ruhe の Spectral Shift hanczos (1980 Math. Comp)[4]

が相次いであらわれた.* これらはそのまゝでは汎用性に問題があり、メーカーのプログラムプロダクトとするには尚多くの検討，改造を要すると推測されるが、両者を統合してやれば、汎用性は兎も角として、可なり強力なものとなし得るのではないかと思われた。汎用性に関しては、Spectral Shift の戦略に問題ありと感じられるので、'兎も角' と言ったわけである。

ところで、前報：スツルム・同時逆反復法のプログラム，GINVRY[1]において開発したグループ分け手法は、高度の汎用性があるから、GINVRYのワク組みを基本にして、GINVRYの中の各グループ毎の（同時または単独）逆反復のところを、Parlett·Scott 流の選択的直交化つきランチョスにおきかえたものを作り、それと、GINVRYそのものとの汎用性と性能比較を行って見たいと考えた。

* Parlett が最近書いた本[3]も併読されたい.

2 準備 (2, 3より)

〔exact arith. による場合の主な定理〕

真の固有値を λ_i, ランチョスを j ステップで切って T_j を作り、その固有値を $\theta_i^{(j)}$, $i=1,\cdots,j$ (リッツ値) とする. (j) をしばしば省略して単に θ_i と書く. T_j を固有値解析して $T_j S_j = S_j \Theta_j$ としたとき、これと周知の関係:

$$A Q_j - Q_j T_j = \beta_j q_{j+1} e_j^* \quad (e_j^* = (0,\cdots,\overset{j番目}{\underset{\downarrow}{1}}))$$

とを合せると、残差ベクトル $Ay_i - y_i\theta_i$ (但し $y_i = Q_j s_i$) に関し、固有ベクトル s_i の第 j 要素を s_{ji} と書くとき、

$$Ay_i - y_i\theta_i = AQ_j s_i - Q_j s_i \theta_i = \beta_j q_{j+1} e_j^* s_i ,$$

これから、

---

〔1〕 $\quad \| Ay_i - y_i\theta_i \| = \beta_j \| e_j^* s_i \| = \beta_j |s_{ji}|$

(これは有用な関係である. プログラムの中でも使う)

次いで、残差ノルムと近似固有値についての関係から、

---

〔2〕 各 θ_i に対し $\exists i'$; $|\lambda_{i'} - \theta_i| \leq \beta_j \| e_j^* s_i \| = \beta_j |s_{ji}|$

これより積極的に (固有値の近接度と関係させて)

---

〔3〕 $A z_{i'} = \lambda_{i'} z_{i'}$, $\gamma_i = \min_{k\neq i'} |\lambda_k - \theta_i|$ として、$\exists i'$;
$|\lambda_{i'} - \theta_i| \leq (\beta_j |s_{ji}|)^2 / \gamma_i$

γ_i は計算できないが、代りに $\delta_i = \min_{k\neq i} |(\theta_k \pm \beta_j |s_{ji}|) - \theta_i|$ がゼロでないならば、γ_i のところに δ_i を代用できる. こうして、プログラムの中でも利用できるが、今回は使わなかった.

さてコンピュータ演算となると、上記のようには行かない。まず直交関係 $Q_j^* Q_j = I$ が崩れる。$Q_j^* Q_j$ の最小固有値の平方根を $\sigma_1(Q)$ とすると〔2〕は次の形をとる：

〔4〕 $|\lambda_i - \theta_i| \leqslant \sqrt{2}\,(\beta_j |e_j^* s_i| + \|F_j\|)/\sigma_1(Q_j)$,

ここに F_j は、 $AQ_j - Q_j T_j = \beta_j q_{j+1} e_j^* - F_j$ （丸め誤差によるもの）

$\|F_j\|$ は、$n*$(計算機eps)$*\|A\|$ のオーダ以下ゆえそう心配することはないが、$\sigma_1(Q_j)$ が1から離れて殆んどゼロになることがある。これが困る。次のPaigeの定理が、選択的直交化っきランチョスの基礎となっている：

〔5-1〕 $y_i^* y_k = \left[g_{ii}(s_{jk}/s_{ji}) - g_{kk}(s_{ji}/s_{jk}) + f_{ik} \right] / (\theta_i - \theta_k)$

〔5-2〕 $y_i^* q_{j+1} = g_{ii}/\beta_{ji} = g_{ii}/(\beta_j |s_{ji}|)$

ここに G, F は丸め誤差行列で、$\|G\| \fallingdotseq \|F\| \fallingdotseq \sqrt{n}*$(計算機eps)$\|T_j\|$.

〔5-1〕：收束していない リッツベクトル y_i, y_k はよく直交しており、一方，收束していない y_k には、よく收束している y_i の成分を多く含むと主張．

〔5-2〕：q_{j+1} には、よく收束している y_i の成分を多く含むと主張．

そこで、q_{j+1} から、よく收束している y_i の成分をぬきとるという操作を各 j についてやって行けば、それで十分になるのではないか？．これがParlett等の考え方の大筋である*．今回この線に沿ってプログラムを作成した。それを次に示す。

*(2) には、他にいくつかの戦略が併記されているが、今回は(3)にのっているものに大筋を従った．

3 テストに使用のプログラムの体系とLANSSO

{ $Ax=\lambda x$ の $0<\lambda_i<al$ なる λ_i と v_i $(i=1,\cdots,nc)$ を求める }

eps, epso, epsl, epsq を与える. (例 10^{-9}, $0.2*10^{-9}$, 10^{-6}, 10^{-3})

SSTRM使用の二分法 BISECT により、nc と粗いグループ分け情報 nc[i], xu[i], xl[i] , i=1,…,nc を得る.

合理的なグループ分けを行う、SSTRM を用び使用
グループ番号 ig , 固有値個数 ln[ig],
グループ下限 xug[ig], 上限 xlg[ig] , igmax を決定

k=1
do ig = 1, igmax　　{ここまではGINVRY(1)と同じ}

h = (xlg[ig]-xug[ig])/2.0
shift = xug[ig]+h　　{ig区間の中央}

① B = A - shift*I をコレスキ U^TDU

r = (DU)$^{-1}$e　{ウイルキンソンの初期ベクトル}

② r = B^{-1}r

LANSSO { LANczos with Selective Orth. }
(次頁)

400 CONTINUE　　{ここから先はGINVRYと同じ}

LANSSO で収束しないもの : difi > eps を RL商シフトつき逆反復*で止めを刺す

k = k+ln[ig]　{kを次のグループの先頭に進める}

① : Shift値が異常に固有値に近い, その他の理由で|ピボット値|がある程度以下になったときは、shift値を移動してやり直す. (GINVRYと同じ)

② : 1回逆反復をほどこして shift値から遠い(グループの外の)固有値に対する固有ベクトル成分を予じめふるっておく. (結果が若干よい様る)

* Rayleigh 商を毎回作ってシフト実を更新しながら行なう逆反復, RL逆反復と略称.

LANSSO の主要部

```
                                                    {これよりランチョス}
      q1 = r ; u = B⁻¹r ; α = q1ᵀu
      r = u − α*q1 ; β = (rᵀr)^(1/2)
      ta[1] = α ; tb[1] = β ; norm = |α|+|β|
      C = −1
③     jmax = 16 + 8*ln[ig]
      do j = 2, jmax
          count = 0
          q = r/β ; u = B⁻¹q ; r = u − β*q_(j−1) ; α = qᵀr
          r = r − α*q ; β = (rᵀr)^(1/2) ; q_j = q
          ta[j] = α ; tb[j] = β ; norm = max(norm, |α|+2|β|)
④         C = C*(−1)
          if C=1 then
              三重対角行列 Tj の固有値の逆数 μji (i=1,…,j) を求め、それを絶対値の小さい方から順にならべ、改めて μji とする。
⑤             −h−epsq < μji < +h+epsq なる i の max を nqd とする.
              difom = 0.0
⑤             do i = 1, nqd    {これより Selective Orth.}
                  Tj の固有ベクトル Si = (s1i,…,sji)ᵀ を計算
                  bji = |tb[j]|*|sji| ; epsqn = epsq*norm
⑥                 if bji < epsqn then
                      yi = Qj Si ; λi = shift + μji
                      r から yi 成分をぬきとる.
                      difo = ‖A yi − λi yi‖
                      epsc = epsl*norm ; difom = max(difo, difom)
⑦                     if (difo<epsc) ∧ (−h≤μji≤h) count = count+1
⑧                 if (count ≥ ln[ig]) ∧ ((j > L*(ln[ig]+2)) ∨ (difo < epso))
                      go to 400
⑨     if count < ln[ig] ifusok = ln[ig] − count
400   continue
```

③：例えば、グループ内の λ_i の個数 $ln[ig]$ が 5 なら $jmax = 16+40 = 56$.
まだ、これで本当によいかどうか確かでない.

④：毎回 T_j を固有値解析し、再直交化するのは馬鹿らしいからこうした.

⑤：ngd は number of good vectors のつもり.

⑥：'$bij < epsqn$' であるような ベクトル s_i に対しては、いわゆる リッツ ベクトル（もとの空間にひきのばされた $y_i = Q_j s_i$）を作り、Ⅱ からその成分をぬきとる. ここが Parlett の LANSO のアイディアである.

⑦：これで、また一本 解ベクトルを捕捉したというわけである.

⑧：この if 文は大変こみ入っているが、非常に重要である。（要するに、どこらあたりで LANSSO に見切りをつけて、最後の RL 商にフトつき逆反復にバトンを渡すのが適当か、というわけである.） 特に条件：

$$j > L*(ln[ig]+2)$$

のためのパラメータ L は、GINVRY の経験からとりあえず決めているが、まだ自信はない. 今のところ、L は、epso 値と関係させて、

$epso = 10^{-9} \sim 10^{-12}$ のときは $L=4$
$epso = 10^{-12} \sim 10^{-15}$ のときは $L=5$

と考えているが今回はすべて $L=4$ のときのデータを載せた.

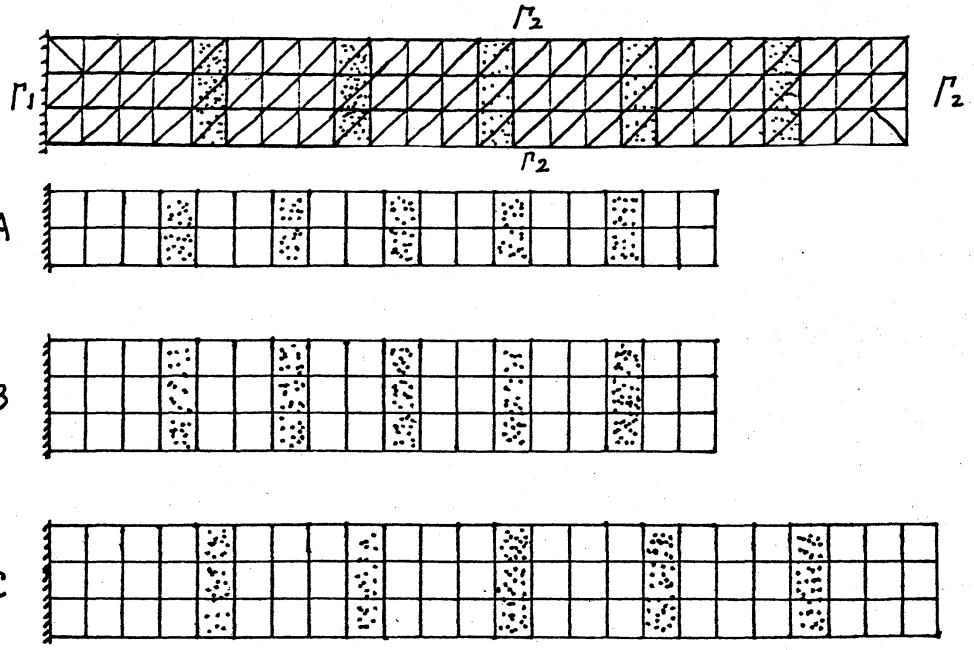
⑨ $count < ln[ig]$ のことが、epsbi の最適値使用のときは、A1, A2, A4, B1, B4, C1, C4 の諸ケースで、$epso = 0.2*10^{-14}$ までの範囲のテストケースで一度も起っていないが epsbi を故意に最適値の値から大きく外すと、盛に起る.

4 テスト問題とテストの現状

テスト問題として、とりあえずGINVRY[1]に使ったものの中から、A1, C1（近接度の強い固有値をもつものの代表）及びA4, B4, C4（素直な固有値分布をしているものの代表）をえらんだ。

支配方程式は、$\nabla\cdot(-k(x)\nabla u)=\lambda u \Rightarrow A\underset{\sim}{u}=\lambda\underset{\sim}{u}$

Γ_1：固定境界条件 $u=0$ ，　Γ_2：自由境界条件 $\nabla u=0$



上図で、□部の k を1.0とし、▨部の k を左から、（X=A,B,Cとして）

問題X1	10^{-2}	10^{-5}	10^{-5}	10^{-8}	10^{-8}
問題X2	10^{-1}	10^{-3}	10^{-3}	10^{-5}	10^{-5}
問題X3	1.0^{-1}	10^{-2}	10^{-2}	10^{-3}	10^{-3}
問題X4	0.31	10^{-1}	1.0^{-1}	10^{-2}	10^{-2}

例えば"Aの場で物質定数 k を問題1のようにえらんだとき、問題A1と略称する。

テストの現状：

epsbi値としては、0.4，0.2，0.1 の三ケースについて行なった。このテストに使用したRL逆反復のプログラムは、GINVRYに対してはこのテスト範囲で問題を起していなかったのであるが、LANSSOの後につなぐときには問題A1の次のケスで問題を起した．

epsbi = 0.4 ,　　eps = 10^{-14} , 10^{-12} , 10^{-9}

epsbi = 0.2 ,　　eps = 10^{-14}

epsbi = 0.1 ,　　eps = 10^{-14}

問題 A1 の

0.633,974,597,5…, 0.633,974,599,3…, 0.633,974,604,3…, 0.633,974,604,9…

という 4個の近接固有値の中の2個（一線）がRL逆反復過程で滞留して所望精度まで行くまえに反復5回打ちきりにかゝるのである。テストの当時には、A1のような甚だしい近接固有値をもつ問題に対しては無理と誤認して、テストの主力をA4，B4，C4の系列に移してしまった。その後GINVRYでも、epsbi その他を故意に適値から大中に外すとき、同様の滞留現象が稀には起ることが判り、原因はRL逆反復に使用したコレスキにあることがつきとめられた。しかし、改良されたRL逆反復によってLANSSOの方を再テストしていないので、今回の報告は非常に不完全なものである。

上述の理由により、問題の素直な A4, B4, C4 の系列についての両者の比較を主として示そう.

$m=160$ に換算した CPUカウントと USEカウントを示す.

(スツルム・同時逆反復法[1]におけると同じ換算法をとる.)

問題 A4 $eps=10^{-9}$, $al=1.5$ ($nc=23$)　　最大直交化本数 ↓

epsbi	共通部		(LQM=6) SUBSPACE				: TOTAL		(L=4) LANSSO				: TOTAL		
	SSTRM	LU	反復	4P:V:/m	P:V:/m	RL	CPU	USE	反復	4V:/m	V:/m	RL	CPU	USE	
0.4	10	4	84	15	4	6	35	98	90	2	1	0	16	95	14
0.2	15	6	88	12	3	6	39	103	92	2	1	2	25	101	14
						平均→	37	101					21	98	

問題 B4 $eps=10^{-9}$, $al=1.5$ ($nc=29$)

			(LQM=6)						(L=4)						
0.4	10	3	85	25	6	7	45	101	81	2	1	0	15	85	23
0.2	17	6	107	17	4	5	43	122	122	3	1	1	27	130	17
							44	112					21	113	

問題 C4 $eps=10^{-9}$, $al=1.5$ ($nc=29$)

			(LQM=6)						(L=4)						
0.4	10	4	104	25	6	6	45	120	94	2	1	1	17	100	18
0.2	15	6	84	12	3	3	36	96	110	3	1	1	25	118	19
							41	108					22	108	

$eps=10^{-9}$ のとき、

CPUカウントについては LANSSO 使用の方が SUBSPACE 使用のものとくらべて約半分, USEカウントについてはほぼ同じである. ($m=80$ で評価すると、LANSSO の方がもっと有利になる.) 次に $eps=10^{-14}$ としたときどうなるかを示そう:

問題 A4　eps = 10^{-14}　al = 1.5　(nc = 23)

0.4	10	4	92	15	4	18	47	118	128	3	1	8	25	141	28
0.2	15	6	100	12	3	9	42	118	146	4	1	8	33	161	23
							45	116					29	156	

問題 B4　eps = 10^{-14}　al = 1.5　(nc = 29)

0.4	10	3	98	29	7	13	55	121	143	4	1	21	39	168	23
0.2	17	6	120	18	5	13	54	144	170	4	1	19	46	196	29
							55	133					43	182	

問題 C4　eps = 10^{-14}　al = 1.5　(nc = 29)

0.4	10	4	108	25	6	10	49	128	152	4	1	22	40	179	31
0.2	15	6	120	18	5	14	53	145	170	4	1	8	33	185	28
							51	137					37	182	

eps = 10^{-14} においても、CPUカウントについてはLANSSO有利，USEカウントについてはLANSSO不利となっているがこの程度のことなら、何れにしても大差はないと言うこともできよう。

約言して、‘LANSSOは、あまり甚だしい近接固有値をもたない素直な問題を、精度要求を極限まで追求するのでなく解きたいというときには大変有望である’と言えようか。

問題A1，C1の系列については先程述べたような歯切れの悪い事情があるが、C1についてはLANSSOも正常に動作した

ので、それから先に比較しよう.

問題 C1 $eps = 10^{-9}$ $al = 1.5$ $(nc = 30)$

epsbi	共通部		LQM=5 SUBSPACE					Total	L=4 LANSSO					Total	
	SSTRH	LU	反復			RL	CPU	USE				RL	CPU	USE	
0.4	9	4	60	12	3	2	27	69	138	3	1	5	21	143	17
0.2	13	6	61	7	2	3	29	72	122	3	1	1	23	132	14
0.5	16	7	71	8	2	3	34	83	153	4	1	2	29	168	32
							30	75					23	146	

問題 C1 $eps = 10^{-14}$ $al = 1.5$ $(nc = 30)$

			LQM=5						L=4						
	9	4	95	25	6	7	45	112	156	4	1	22	39	183	29
	13	6	97	15	4	4	38	111	174	4	1	8	31	189	32
	16	7	91	11	3	8	42	109	183	5	1	25	53	216	32
							42	111					41	196	

以上 C1 においては $eps = 10^{-14}$ のとき CPUカウントは同じぐらい、USEカウントはLANSSOの方が2倍に及ぶ。

問題A1 についてのデータには問題があるが、一応あげる.

問題A1 $eps = 10^{-9}$ $al = 2.0$ $(nc = 30)$

	14	5	72	11	3	1	31	81	129	3	1	14* (8)	36* (30)	149	13
	20	7	76	10	3	0	37	86	117	3	1	0	30	125	14
	24	9	88	7	2	1	41	100	119	3	1	0	26	129	16
							36	89					34* (32)	134	

* 2ケ滞留

問題A1 $eps = 10^{-14}$ $al = 2.0$ $(nc = 30)$

	14	5	90	15	3	8	42	107	161	4	1	31* (25)	54** (48)	206	2
	20	7	97	13	3	1	41	108	163	4	1	35* (23)	66* (54)	205	3
	24	9	102	10	3	5	48	118	175	4	1	28* (16)	64* (52)	183	1
							44	111					55* (51)	191	

* 3ケ滞留

＊印のものはRL逆反復が滞留して反復最大値6までで打ちきりになったものがあることを示す．A1，$eps=10^{-9}$, $epsbi=0.4$のとき2個，A1 $eps=10^{-14}$のときはそれぞれ3個ある．RL逆反復のプログラムを改良すれば、それぞれのdifoの内容からみて()内の値で収束すると考えている。とすると、A1においては$eps=10^{-14}$のときCPUカウントはLANSSOの方が若干多い程度でありUSEカウントはLANSSOが約2倍程度というイメージとなろう。

結局約言すれば、‘LANSSO使用のスツルム・ランチョス・RL逆反復法は、甚だしい密集固有値をもつ問題を計算機の精度の限界近くまでの精度で求めようとするときには問題であるが、そうでない問題領域で、スツルム・逆反復（同時）・RL逆反復よりも有利な分野が、可なりありそうである’と言えようか、しかし、LANSSOは、すでに述べたこと以外にも、テスト段階でいろいろデリケートな振舞に出会っているので、汎用化のためには、これから尚多くの工数を必要としよう。

なほ、問題A1の場合，Parlett流の選択的用直交化なしでは全く使えなかったこと、また30個の固有値を一度にやらせようとしたときには、LANSSOでもうまく行かなかったことを附言しておきたい．

文献

(1) 村田，後 'スツルム・同時逆反復法' 本講究録

(2) Parlett · Scott ' Lanczos with Selective Orth' (1979, Math. Comp.)

(3) Parlett 'The Symmetric Eigenvalue Problems' (1980 Prentice H.)

(4) Ericcson · Ruhe ' Spectral Shift Lanczos' (1980 Math. Comp.)